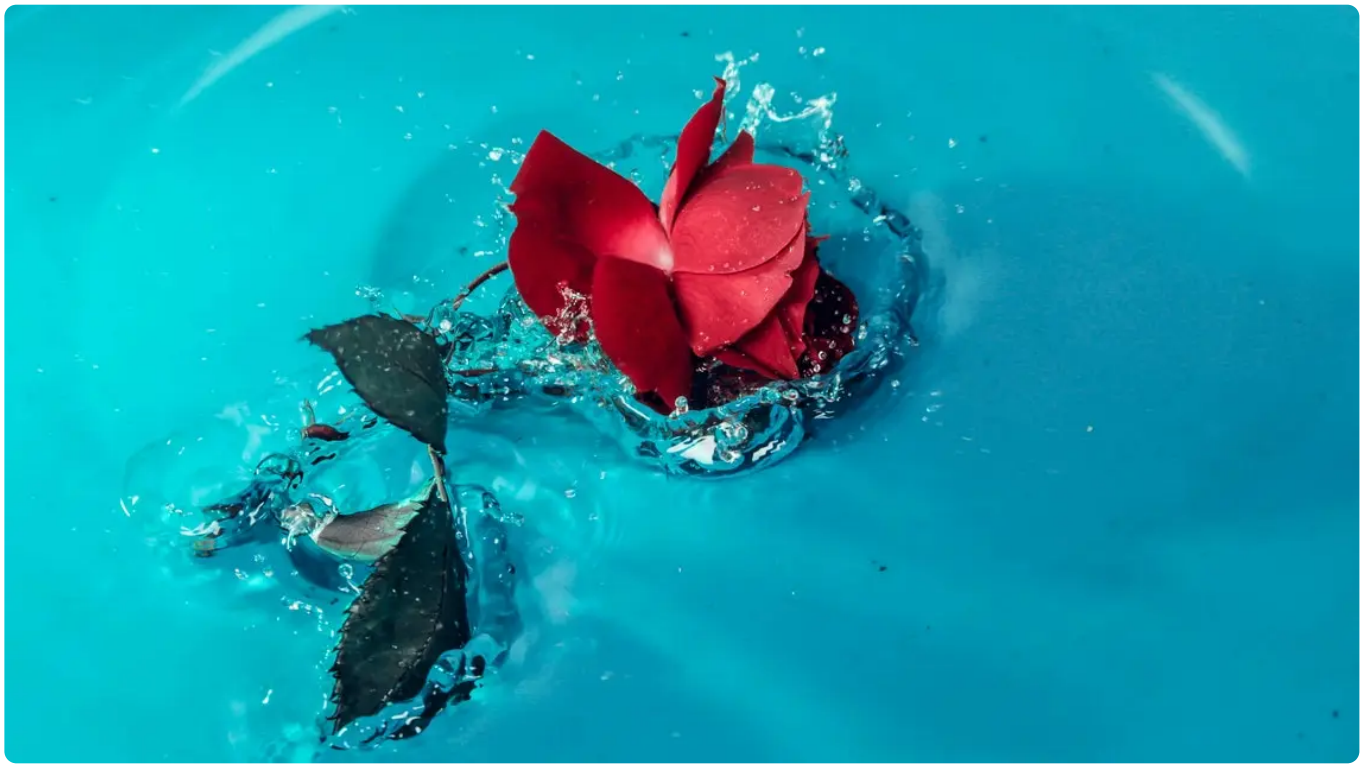
下载APP

03 | 实战5步（上）：怎么定义问题和预处理数据？

2021-09-03 黄佳

《零基础实战机器学习》 课程介绍 >

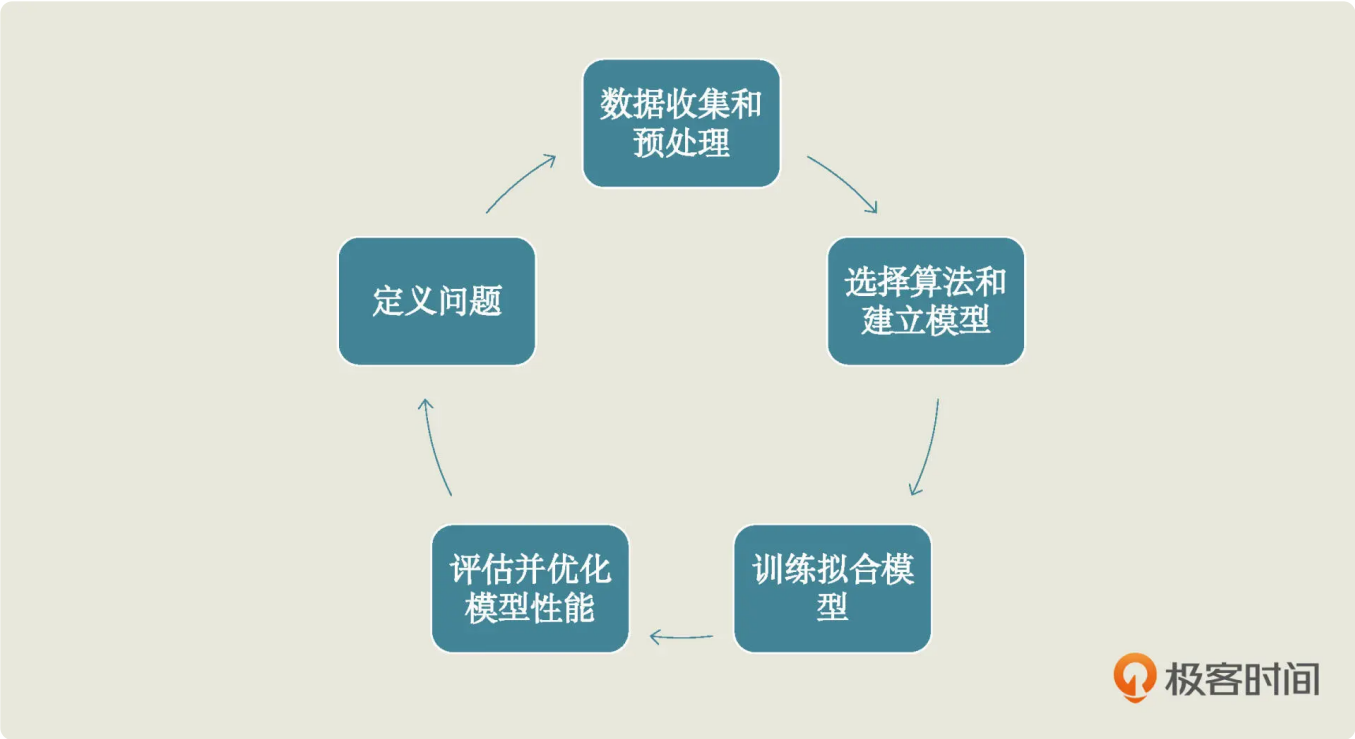


讲述：黄佳
时长 21:38 大小 19.81M

你好，我是黄佳。

在《打好基础：到底什么是机器学习？》中，我和你说了到底什么是机器学习，你还记得我们的结论吗？机器学习是一种从数据生成规则、发现模型，来帮助我们预测、判断、分析和解决问题的技术。现在，你是不是跃跃欲试，准备动手开跑机器学习程序了？

不要着急，在实战之前你还需要掌握最后一个知识点，也就是机器学习项目分为哪些步骤，你才好跟着一步步走。一个机器学习项目从开始到结束大致分为 5 步，分别是**定义问题、收集数据和预处理、选择算法和确定模型、训练拟合模型、评估并优化模型性能。**5 步是一个循环迭代的过程，你可以参考下面的图片：



我们所有的项目都会按照这 5 步来做，我把它简称为实战 5 步。为了让你更深地理解这 5 步，在后面能更快地上手，我会带你做一个项目，我会给你清楚解释每一个步骤的目的和背后的原理。我会把这个项目分成两节课，今天我们先来一起搞定前两步，也就是**定义问题和数据的预处理。**

好了，我们正式开始吧！

第 1 步 定义问题

我们先来看第一步，定义问题。在定义问题这个环节中，我们要剖析业务场景，设定清晰的目标，同时还要明确当前问题属于哪一种机器学习类型。如果不搞清楚这些，我们后面就无法选择模型了。

所以首先，我们先得来了解一下我们这个项目的业务场景。假设你已经入职了“易速鲜花”的运营部，正在对微信公众号推广文案的运营效率进行分析。你收集了大量的软文数据，包括点赞数、转发数和浏览量等等，就像下面这样：

	A	B	C	D	E
1	点赞数	转发数	热度指数	文章评级	浏览量
2	2646	1347	7	5	260004
3	816	816	4	6	100004
4	1224	612	6	5	164502
5	1261	1261	6	6	163001
6	1720	1720	7	5	260401
7	1541	1541	7	5	220002

因为微信公众号阅读量超过 10 万之后，就不能显示它的具体阅读量了。所以针对这个问题，我们项目的目标就是，建立一个机器学习模型，根据点赞数和转发数等指标，估计一篇文章能实现多大的浏览量。

因为要估计浏览量，所以在这个数据集中：点赞数、转发数、热度指数、文章评级，这 4 个字段都是特征，浏览量就是标签。这里我们已经有要估计的标签了，所以这是一个监督学习问题。再加上我们的标签是连续性的数值，因此它是一个回归问题。

不难看出，在这个数据集中，特征和标签之间明显呈现出一种相关性：点赞数、转发数多的时候，往往浏览量也多。但是，这种相关性可以通过哪个具体的函数来描绘呢？目前我们还不知道，所以我们在这个项目中的任务就是找到这个函数。

这样，我们就已经把问题定义好了，紧接着下一步就是数据收集和预处理。

第 2 步 收集数据和预处理

“数据的收集和预处理”在所有机器学习项目中都会出现，它的作用是为机器学习模型提供好的燃料。数据好，模型才跑得更带劲。这步骤看似只有一句话，其实里面包含了好几个小步骤，完整来讲有 6 步：

收集数据；

数据可视化；

数据清洗；

特征工程；

构建特征集和标签集；

拆分训练集、验证集和测试集。

你可能一眼看上去不太明白这 6 个步骤的意思。不要着急，接下来我会继续结合“易速鲜花”这个项目挨个解释的。

1. 收集数据

首先是收集数据，这一步又叫采集数据。在我们的项目中，我已经把它做好了，你可以在这里下载现成的数据集。

不过，在现实中，收集数据通常很辛苦，要在运营环节做很多数据埋点、获取用户消费等行为信息和兴趣偏好信息，有时候还需要上网爬取数据。收集数据不是我们的课程重点，你有兴趣的话我向你推荐陈旸老师的《数据分析实战 45 讲》，里面有很多收集数据的方法。

有了数据集，接下来我们要做的是数据可视化的工作，也就是通过可视化去观察一下数据，为选择具体的机器学习模型找找感觉。

2. 数据可视化

数据可视化是个万金油技能，能做的事非常多。比如说，可以看一看特征和标签之间可能存在的关系，也可以看看数据里有没有“脏数据”和“离群点”等。

不过在正式可视化之前，我们需要把收集到的数据导入运行环境。数据导入我们需要用到 Pandas 数据处理工具包。这个包可是操作数据的利器，我们未来每个项目中都会用到。现在我们用 import 语句导入它：

复制代码

```
import pandas as pd # 导入Pandas数据处理工具包
```

然后，我们通过下面的代码把这个项目的数据集读入到 Python 运行环境，用 DataFrame 的形式呈现出来：

复制代码

```
df_ads = pd.read_csv('易速鲜花微信软文.csv') # 读入数据
df_ads.head() # 显示前几行数据
```

DataFrame 是机器学习中常见的二维表格类型数据结构。在上面的代码中，我用 read_csv API 把 CSV 格式的数据文件，读入到 Pandas 的 DataFrame 中，把它命名为了 df_ads。这段代码输出如下：

	点赞数	转发数	热度指数	文章评级	浏览量
0	2646	1347.0	7	5	260004
1	816	816.0	4	6	100004
2	1224	612.0	6	5	164502
3	1261	1261.0	6	6	163001
4	1720	1720.0	7	5	260401
5	1541	1541.0	7	5	220002
6	827	NaN	5	7	107503
7	866	686.0	5	9	110504
8	1314	1300.0	8	5	229461
9	1431	1099.0	6	5	165004

这样就完成了数据的导入了，接着我们就可以正式进入“可视化”了。根据经验，我们猜测“点赞数”最有可能和“浏览量”之间存在线性关系。那是不是真的这样呢？我们可以画出图来验证一下。

在这个“验证”环节中，我们需要用到两个包：一个是 Python 画图工具库“Matplotlib”，另一个是统计学数据可视化工具库“Seaborn”。这两个包都是 Python 数据可视化的必备工具包，它们是 Anaconda 默认安装包的一部分，不需要 pip install 语句重复安装。

在导入这两个包时，我们依旧用 import 语句。请你注意，为了节省代码量，我并没有导入完整的 matplotlib 包，而是只导入了 matplotlib 包中的绘图模块 pyplot：

复制代码

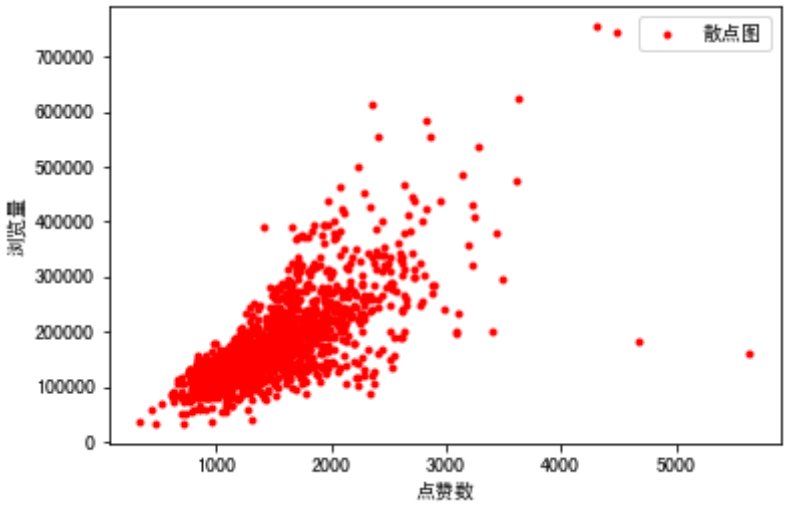
```
#导入数据可视化所需要的库
import matplotlib.pyplot as plt # Matplotlib - Python画图工具库
import seaborn as sns # Seaborn - 统计学数据可视化工具库
```

因为线性关系可以简单地用散点图来验证一下。所以下面我们用 matplotlib 包中的 plot API，绘制出“点赞数”和“浏览量”之间的散点图，看看它们的分布状态。

复制代码

```
plt.plot(df_ads['点赞数'],df_ads['浏览量'],'r.', label='Training data') # 用matp
plt.xlabel('点赞数') # x轴Label
plt.ylabel('浏览量') # y轴Label
plt.legend() # 显示图例
plt.show() # 显示绘图结果！
```

输出的结果如下图所示：



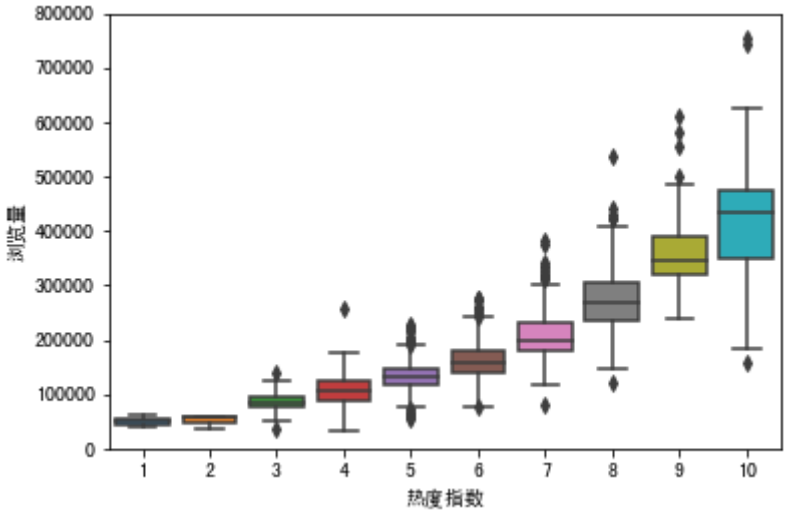
从这张图中我们可以看出来，这些数据基本上集中在一条线附近，所以它的标签和特征之间，好像真的存在着线性的关系，这可以为我们将来选模型提供参考信息。

接下来，我要用 Seaborn 的 boxplot 工具画个箱线图。来看看这个数据集里有没有“离群点”。我这里随便选择了热度指数这个特征，你也可以为其它特征试试绘制箱线图。

复制代码

```
data = pd.concat([df_ads['浏览量'], df_ads['热度指数']], axis=1) # 浏览量和热度指数
fig = sns.boxplot(x='热度指数', y="浏览量", data=data) # 用seaborn的箱线图画图
fig.axis(ymin=0, ymax=800000); #设定y轴坐标
```

下图就是我们输出的箱线图：



箱线图是由五个数值点组成，分别是最小值 (min)、下四分位数 (Q1)、中位数 (median)、上四分位数 (Q3) 和最大值 (max)。在统计学上，这叫做五数概括。这五个数值可以清楚地为我们展示数据的分布和离散程度。

这个图中下四分位数、中位数、上四分位数组成一个“带有隔间的盒子”，就是所谓的箱；上四分位数到最大值之间建立一条延伸线，就是所谓的线，也叫“胡须”；胡须的两极就是最小值与最大值；此外，箱线图还会将离群的数据点单独绘出。

在上面这个箱线图中，我们不难发现，热度指数越高，浏览量的中位数越大。我们还可以看到，有一些离群的数据点，比其它的文章浏览量大了很多，这些“离群点”就是我们说的“爆款文章”了。

到这里，数据可视化工作算是基本完成了。在数据可视化之后，下一步就是数据的清洗。

3. 数据清洗

很多人都把数据清洗比作“炒菜”前的“洗菜”，也就是说数据越干净，模型的效果也就越好。清洗的数据一般分为 4 种情况：

第一种是处理缺失的数据。如果备份系统里面有缺了的数据，那我们尽量补录；如果没有，我们可以剔除掉残缺的数据，也可以用其他数据记录的平均值、随机值或者 0 值来补值。这个补值的过程叫数据修复。

第二个是处理重复的数据：如果是完全相同的重复数据处理，删掉就行了。可如果同一个主键出现两行不同的数据，比如同一个身份证号后面有两条不同的地址，我们就要看看有

没有其他辅助的信息可以帮助我们判断（如时戳），要是无法判断的话，只能随机删除或者全部保留。

第三个是处理错误的数据：比如商品的销售量、销售金额出现负值，这时候就需要删除或者转成有意义的正值。再比如表示百分比或概率的字段，如果值大于 1，也属于逻辑错误数据。

第四个是处理不可用的数据：这指的是整理数据的格式，比如有些商品以人民币为单位，有些以美元为单位，就需要先统一。另一个常见例子是把“是”、“否”转换成“1”、“0”值再输入机器学习模型。

那么如何看数据集中有没有脏数据呢？

就我们这个项目的数据集来说，细心的你可能在 DataFrame 图中已经发现，行索引为 6 的数据中“转发数”的值是“NaN”，意思是 Not A Number。在 Python 中，它代表无法表示、也无法处理的值。这是典型的脏数据。

	点赞数	转发数	热度指数	文章评级	浏览量
0	2646	1347.0	7	5	260004
1	816	816.0	4	6	100004
2	1224	612.0	6	5	164502
3	1261	1261.0	6	6	163001
4	1720	1720.0	7	5	260401
5	1541	1541.0	7	5	220002
6	827	NaN	5	7	107503
7	866	686.0	5	9	110504
8	1314	1300.0	8	5	229461
9	1431	1099.0	6	5	165004

我们可以通过 DataFrame 的 isna().sum() 函数来统计所有的 NaN 的个数。这样，我们就可以在看看有没有 NaN 的同时，也看看 NaN 出现的次数。如果 NaN 过多的话，那么说明这个数据集质量不好，就要找找数据源出了什么问题。

复制代码

```
1 df_ads.isna().sum() # NaN出现的次数
```

输出结果如下:

复制代码

```
点赞数      0
转发数     37
热度指数     0
文章评级     0
浏览量      0
dtype: int64
```

输出显示我们的数据集中“转发数”这个字段有 37 个 NaN 值。对于上千条数据的数据集，这还不算很多。那么该如何处理呢？也很简单。我们可以用 dropna() 这个 API 把出现了 NaN 的数据行删掉。

复制代码

```
df_ads = df_ads.dropna() # 把出现了NaN的数据行删掉
```

你可能会觉得，我们刚才通过箱线图找到了离群点（爆款文章），这些算是脏数据吗？这是个很好的问题，而且这个问题并没有固定的答案。

删去了离群点，模型针对普通的数据会拟合得比较漂亮。但是现实生活中，就是存在着这样的离群点，让模型不那么漂亮。如果把这里的离群点删掉，那模型就不能工作得那么好了。所以，这是一个平衡和取舍的过程。

我们可以训练出包含这些离群点的模型，以及不包含这些离群点的模型，并进行比较。在这里，我建议保留这些“离群点”。

现在，我们就完成了对这个数据的简单清洗。不同类型的数据有不同的清洗方法，我们这里就不一一介绍了。后续项目中，针对具体项目和数据集，我们还会再细讲的。我们继续讲下一个步骤，把数据转换成机器所更易于读取的格式，也就是特征工程。

4. 特征工程

特征工程是一个专门的机器学习子领域，而且我认为它是数据处理过程中**最有创造力的环节**，特征工程做得好不好，非常影响机器学习模型的效率。

我举个例子来解释下什么是特征工程。你知道什么是 BMI 指数吗？它等于体重除以身高的平方，这就是一个特征工程。

什么意思呢？就是说经过了这个过程，BMI 这一个指数就替代了原来的两个特征——体重和身高，而且完全能客观地描绘我们的身材情况。

因此，经过了这个特征工程，我们可以把 BIM 指数作为新特征，输入用于评估健康情况的机器学习模型。

你可能会问这样做的好处是什么？以 BMI 特征工程为例，它降低了特征数据集的维度。维度就是数据集特征的个数。要知道，在数据集中，每多一个特征，模型拟合时的特征空间就更大，运算量也就更大。所以，**摒弃掉冗余的特征、降低特征的维度，能使机器学习模型训练得更快。**

这只是特征工程的诸多的妙处之一，此外特征工程还能更好地表示业务逻辑，并提升机器学习模型的性能。

由于我们这个项目的问题相对简单，对特征工程的要求并不高，这里暂时不做特征工程。后面我会用一节课专门讲解各类特征工程的应用。下面，我们直接进入下一个子步骤，也就是构建机器学习的特征集和标签集。

5. 构建特征集和标签集

我们说过，特征就是所收集的各个数据点，是要输入机器学习模型的变量。而标签是要预测、判断或者分类的内容。对于所有监督学习算法，我们都需要向模型中输入“特征集”和“标签集”这两组数据。因此，在开始机器学习的模型搭建之前，我们需要先构建一个特征数据集和一个标签数据集。

具体的构建过程也很简单，我们只要从原始数据集删除我们不需要的数据就行了。在这个项目中，特征是点赞数、转发数、热度指数和文章评级，所以只需要从原始数据集中删除“浏览量”就行了。

复制代码

```
1 X = df_ads.drop(['浏览量'],axis=1) # 特征集，Drop掉标签相关字段
```

而标签是我们想要预测的浏览量，因此，我们在标签数据集中只保留“浏览量”字段：

```
1 y = df_ads.浏览量 # 标签集
```

复制代码

下面我们再看看特征集和标签集里面都有什么数据。

```
1 X.head() # 显示前几行数据
```

复制代码

```
1 y.head() #显示前几行数据
```

复制代码

因为 Notebook 一个单元格只能有一个输出。所以我把显示两个数据的代码放在了不同的单元格中。它们的输出结果如下图所示：

	点赞数	转发数	热度指数	文章评级
0	2646	1347	7	5
1	816	816	4	6
2	1224	612	6	5
3	1261	1261	6	6
4	1720	1720	7	5

```
0    260004
1    100004
2    164502
3    163001
4    260401
Name: 浏览量, dtype: int64
```

可以看到，除了浏览量之外，所有其它字段仍然都在特征数据集中，而只有浏览量被保存在了标签数据集中，也就是说原始数据集就被拆分成了机器学习的特征集和标签集。

这里我想请你思考的是，无监督学习算法需要这个步骤吗？没错，答案是不需要。因为无监督算法根本就没有标签。

不过，从原数据集从列的维度纵向地拆分成了特征集和标签集后，还需要进一步从行的维度横向拆分。你可能想问，为啥还要拆分呀？因为我们在第一讲《打好基础：弄懂到底什么是机器学习》里就讲过，机器学习并不是通过训练数据集找出一个模型就结束了，我们需要用验证数据集看看这个模型好不好，然后用测试数据集看看模型在新数据上能不能用。

那么具体该怎么拆分呢？这就是我们接下来要解决的问题了。

6. 拆分训练集、验证集和测试集

在拆分之前，我先说明一点，对于学习型项目来说，为了简化流程，经常会省略验证的环节。我们今天的项目比较简单，所以我们也省略了验证，只拆分训练集和测试集，而此时的测试集就肩负着验证和测试双重功能了。

拆分的时候，留作测试的数据比例一般是 20% 或 30%。不过如果你的数据量非常庞大，比如超过 1 百万的时候，那你也不一定非要留这么多。一般来说有上万条的测试数据就足够了。这里我会按照 80/20 的比例来拆分数据。具体的拆分，我们会用机器学习工具包 scikit-learn 里的数据集拆分工具 train_test_split 来完成：

复制代码

```
#将数据集进行80%（训练集）和20%（验证集）的分割
from sklearn.model_selection import train_test_split #导入train_test_split工具
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                    test_size=0.2, random_state=0)
```

这里请你注意一下，虽然是随机分割，但我们要指定一个 random_state 值，这样就保证程序每次运行都分割一样的训练集和测试集。训练集和测试集每次拆分都不一样的话，那比较模型调参前后的优劣就失去了固定的标准。

现在，训练集和测试集拆分也完成了，你会发现原始数据现在变成了四个数据集，分别是：

特征训练集（X_train）

特征测试集（X_test）

标签训练集（y_train）

标签测试集（y_test）

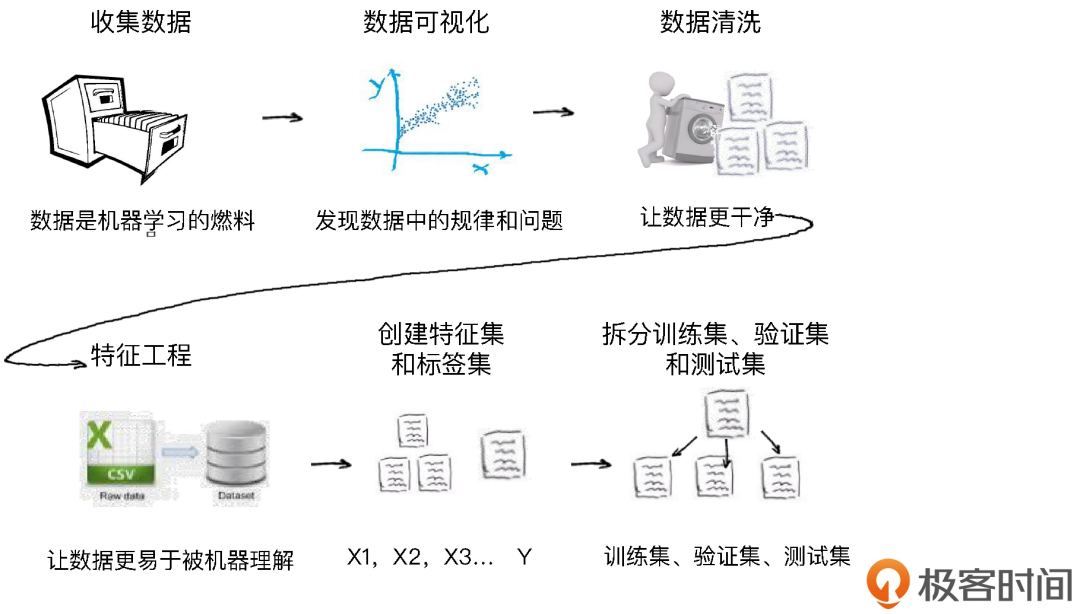
至此，我们全部的数据预处理工作就结束了。

总结一下

现在，我来给你总结一下。

这节课我们介绍了机器学习实战 5 步中的前两步：定义问题以及数据的收集和预处理。不明确定义要解决的问题，我们就没办法有的放矢地选择模型。

而数据的收集和预处理虽然看起来没有模型的选择和优化那么“吸引眼球”，但它其实才是机器学习项目成败的关键。这一步可以分为下图中的 6 个小步：



这 6 步中，尤其是数据可视化和特征工程，因为无定法可循，所以很考验经验，它既是我们对着已有数据找感觉的过程，又是下一步把数据“喂”给模型之前的必要准备。

除此之外，这里我还希望你注意两点：

第一点是，**这些子步骤中的次序并不是固定的**，比如数据可视化和特征工程，很多时候是先可视化，发现了一些特征工程的思路，然后做特征工程，然后再次可视化。而且有的特征工程比如特征缩放，还必须在拆分数据之后做；

第二点，这些子步骤**在一个特定的机器学习项目中，可能不需要全部用到**。比如说无监督学习项目，就不需要创建特征集和标签集这个步骤，一般也不需要用到验证集和测试集。

好的，那么这一讲就到这里，我把这节课使用的代码放在这里了。下一讲，我们要开始选择机器学习模型了，敬请期待！

思考题

这节课到这我们就讲完了，我来给你留两道练习题。

1. 今天我们显示了特征和标签的散点图，还显示了热度指数特征的箱线图。你能不能试着显示一下其它特征之间的散点图，或者箱线图呢？
2. 在数据清洗部分，我们用 dropna() 这个 API，把出现了 NaN 的数据行删掉了。如果你出于保留更多数据的考虑，应该为该字段补值，你会怎么做呢？你能想到哪些补值方法？

提示一种方法：

复制代码

```
df_ads['转发数'].fillna(df_ads['转发数'].mean(), inplace=True) # 用均值补值
```

欢迎你在留言区里分享你做这两个题时的收获和遇到的问题，我在留言区等你。如果这节课帮到了你的话，也欢迎你把这节课分享给自己的朋友。

极客时间

本讲重点

机器学习实战有五大步。

第一步是定义好要解决的问题。
第二步是收集数据和数据的预处理，其中有六个小步骤：

收集数据；
数据清洗；
构建特征集和标签集；
数据可视化；
特征工程；
拆分训练集、验证集和测试集。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

赞 3 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 02 | 工具准备：安装并使用Jupyter Notebook

下一篇 04 | 实战5步（下）：怎么建立估计10万+软文点击率的模型？

精选留言 (8)

写留言

在路上 置顶
2021-09-03

黄老师好，面对空缺的数据，可以采用默认值、均值、中位值等方式填充，有条件的话，可以和业务方沟通，要求补全空缺数据。

展开

作者回复: 谢谢，这是特别好的经验分享。值得置顶！

3

年轻 置顶
2021-09-03

真想一爱看完，太难等了

展开

作者回复: 嗯，谢谢支持。我们这边也多努力一下。

2

谦
2021-09-03

黄老师，请问如果工作人员录入数据的时候空值并没有留空，而是
用“\”，“-”，“无”等字符来标注的话，有什么好的处理方法呢？我一般是用excel进行筛选，查看异常值，然后排序之后吧异常值整行删掉，处理干净之后再用pandas导入。想看看有没有一些更智能的方法。谢谢

展开

作者回复: 有方法。试试这个：
例子：

```
import pandas as pd
data = [[100] ,['/'], ['Zero'], [99]]
columns = ['成绩']
df = pd.DataFrame(data=data, columns=columns)
df
```

原始数据输出：

```
    成绩
0   100
1   /
2   Zero
3   99
```

转换语句

```
df['新成绩'] = pd.to_numeric(df.成绩.astype(str).str.replace(',',''), errors='coerce').fillna(0).astype(int)
df
```

新字段的输出

```
   成绩   新成绩
0  100    100
1  /      0
2  Zero   0
3  99     99
```

是不是比Excel快很多呀？

1 3

茜茜
2021-09-03

谢谢黄老师的讲解。这是我的回答，我还买了老师的书《零基础学机器学习》,昨天浏览了一遍，希望把这本书和专栏的知识都学会。
#思考题1
import matplotlib.pyplot as plt
import seaborn as sns...
展开

作者回复: 感谢支持，一起加油😁！作业完成的非常认真，我看到你还给出了median()和众数的方法mode()。很好。

1

胡了
2021-09-03

老师能催更吗？着急。。。
展开

作者回复: 更新时间：周一，周三，周五😅

1

！null
2021-09-06

从这张图中我们可以看出来，这些数据基本上集中在一条线附近
这是从哪看出来的？
展开

作者回复: 嗯，散点图的那张图，所谓的一条线就是能够穿越散点，使上面的点和下面的点均匀分布在上下的回归线。后面的拟合过程我画出了回归线的变换过程。指得就是那条“想象中的线”。

青松
2021-09-06

真想一次都看完，能否更新快点！

展开

作者回复: 我们加油！

AIK
2021-09-03

想到两个补值方法，我试了感觉都不是很好。第一个是取临近值，这种方法误差很大。第二个是利用对比标签值，反推特征值，这种方法适合小数据量，如果是大数据量很消耗机器性能。还请老师给出个完美的方案。

作者回复: 嗯，非常好。大家都有自己的思考，我也有一个可能的想法，暂时先不分享给大家，让大家继续自由讨论。我过几天在分享这个想法。